



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

SYSTÉM PRO MĚŘENÍ AKUSTICKÝCH VLASTNOSTÍ MÍSTNOSTI

SYSTEM FOR A ROOM ACOUSTICS MEASUREMENT

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MARTIN STOJAN

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. IGOR SZÖKE, Ph.D.

BRNO 2021

Zadání bakalářské práce



Student: **Stojan Martin**
Program: Informační technologie
Název: **Systém pro měření akustických vlastností místností**
System for a Room Acoustics Measurement
Kategorie: Softwarové inženýrství

Zadání:

1. Seznamte se s dodaným HW a SW pro vícekanálové zpracování audia od firmy Audified.
2. Seznamte se s odhadem impulzní odezvy pomocí metody ESS.
3. Seznamte se s procesem měření akustiky místnosti (zaměření mikrofónů, přehrávání měřících signálů, extrakce impulzní odezvy).
4. Navrhněte a implementujte podpůrný webový systém, který poběží na HW (viz bod 1) a usnadní obsluhu celý proces získávání dat a metadat.
5. Celý systém otestujte a získejte zpětnou vazbu od obsluhy. Vyhodnoťte uživatelskou přívětivost.
6. Zhodnoťte dosažené výsledky.
7. Vytvořte A2 plakátek a cca 30 vteřinové video prezentující výsledky vaší práce.

Literatura:

- Dle pokynů vedoucího

Pro udělení zápočtu za první semestr je požadováno:

- Body 1, 2, 3 a část bodu 4 ze zadání.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Szóke Igor, Ing., Ph.D.**
Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2020
Datum odevzdání: 12. května 2021
Datum schválení: 30. října 2020

Abstrakt

Tato práce navazuje na diplomovou práci pana Martina Lacha. Zabývá se systémem na zpracování audia od firmy Audified. Práce pojednává o zpříjemnění ovládání systému, testování funkcí systému s případnými modifikacemi, rozšíření současné funkcionality a vytvoření webové aplikace pro usnadnění zaznamenávání metadat o nahrávání. Také obsahuje návody pro zacházení se systémem a používání zmíněných aplikací.

Abstract

This work follows up on Marin Lach's thesis. It deals with Audified Audio Processing System. The work is about making control of the system more pleasant, testing the functionality of the system with eventual modifications, extending the current functionality and creating a web application to ease the collecting of metadata about the recording. It also contains manuals for dealing with the system and using the mentioned applications.

Klíčová slova

zpracování zvuku, Linux, PHP, JavaScript, Nette, webová aplikace, webový server, metadata

Keywords

audio processing, Linux, PHP, JavaScript, Nette, web application, web server, metadata

Citace

STOJAN, Martin. *Systém pro měření akustických vlastností místnosti*. Brno, 2021. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Igor Szöke, Ph.D.

Systém pro měření akustických vlastností místnosti

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Igora Szökeho, Ph.D.. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Martin Stojan

7. května 2021

Poděkování

Děkuji svému vedoucímu bakalářské práce za zajímavé zadání, vstřícnost, nápomocnost a profesionalitu při konzultacích.

Obsah

1	Úvod	3
2	Popis dodaného HW a SW	4
2.1	Hardware	4
2.2	Boot	6
2.3	Streamer	6
2.4	Receiver	7
2.5	Komunikace mezi Streamerem a Receiverem	8
2.6	Vizualizační a řídicí internetová aplikace	8
3	Analýza problému	10
3.1	Streamer a Receiver	10
3.2	Webová aplikace	11
3.3	Metadata Wizard	11
4	Návod k obsluze	13
4.1	Připojení	13
4.2	Křížová kompilace	13
4.3	Streamer	14
4.4	Receiver	14
4.5	Webová aplikace	16
4.6	Proces zaznamenávání metadat	16
4.7	Aplikace Webwizard	16
5	Webová aplikace Webwizard	18
5.1	Použité technologie	18
5.2	Používání a vzhled aplikace	19
5.2.1	Průchod aplikací	20
5.2.2	Popis jednotlivých stránek	21
5.3	Pohled na programovou část aplikace	24
5.3.1	BasePresenter	24
5.3.2	HomepagePresenter	25
5.3.3	SetupPresenter	25
5.3.4	FinishPresenter	28
5.3.5	EditPresenter	29
5.3.6	PreviousPresenter	29
5.4	Testování aplikace	29
5.4.1	Výstup aplikace	29

5.4.2	Funkcionalita	30
5.5	Známe problémy aplikace	30
6	Závěr	31
	Literatura	32
A	Obsah přiloženého paměťového média	33
A.1	Software	33
A.1.1	Archiv <i>webwizard.zip</i> obsahující vytvořenou webovou aplikaci	33
A.1.2	Archiv <i>zvukovka.zip</i> s upravenými soubory pro nahrávací zařízení . .	33
A.2	Soubor <i>readme</i>	33
A.3	Text Bakalářské práce ve formátu PDF	33
A.4	Archiv se zdrojovým kódem Bakalářské práce	33
A.5	Videoprezentace práce	33
A.6	Plakát	33
B	Manuál	34
B.1	Webwizard	34
B.2	Upravené soubory nahrávacího systému	34

Kapitola 1

Úvod

Práce se zabývá systémem na zpracování audia od firmy Audified. Tento přístroj je pro funkčnost nutné propojit s počítačem. Systém sbírá data z připojených mikrofونů a posílá je počítači k uložení. Tato data se využívají na Fakultě Informačních technologií VUT k dlouhodobému nahrávání ruchů a měření impulsní odezvy místnosti.

Dříve ovládání systému probíhalo konzolovou C++ aplikací. Toto řešení však bylo složité a zdlouhavé, proto byla vytvořena webová aplikace pro pohodlnější užívání systému. Zároveň s nahráváním audio dat je třeba zaznamenávat metadata o prostředí, kde nahrávání probíhá.

K hardware i software tohoto zařízení není existující dokumentace, proto byla orientace v systému zprvu náročná. Velmi nápomocná byla diplomová práce pana Martina Lacha [4], který s tímto systémem pracoval. I přes to však bylo proniknutí do problematiky obtížné. Tato práce proto může posloužit i jako částečný návod pro zprovoznění a chod systému, i pro používání souvisejících programů a aplikací.

Práce navazuje na diplomovou práci pana Martina Lacha, který vytvořil zmíněnou webovou aplikaci pro usnadnění ovládání zvukového systému.

Cílem práce je upravení některých součástí předchozí práce a vytvoření webové aplikace pro pomoc a usnadnění zaznamenávání dat definujících vlastnosti místnosti (její rozměry, materiály stěn) a jednotlivé sady mikrofونů a reproduktorů (pozice, orientace, vlastnosti zařízení). Dalším cílem je kvůli absenci dokumentace a manuálů vytvořit návod jak zacházet s nahrávacím systémem, jeho software a webovými aplikacemi, které s ním souvisejí.

Kapitola 2 se zabývá popisem hardware a software dodaného k práci. Popisuje stavbu HW, průběh bootování, přibližuje aplikace Streamer a Receiver, jejich komunikaci a také webovou aplikaci pro lepší ovládání.

V Kapitole 3 jsou analyzovány obecné i konkrétní problémy systému, záležitost vytváření metadat o prostředí, v němž je systém využíván, a potřeba vytvoření pomocníka se zápisem informací o nahrávání pro další využití.

V Kapitole 4 je vyličen návod pro zacházení se systémem i souvisejícími programy a aplikacemi společně s řešením problémů předchozí implementace.

Kapitola 5 obsahuje popis vytvořené webové aplikace, způsob jejího používání a strukturu.

Kapitolu 6 tvoří závěr práce.

Kapitola 2

Popis dodaného HW a SW

Společně se zadáním práce byl dodán hardware - zvukový systém pro získávání audio dat a několik mikrofونů. Dodán byl také software, pomocí kterého je možné toto zařízení ovládat a získávat nahraná data.

2.1 Hardware

Zvukový systém slouží k nahrávání záznamů z mikrofонů, což lze využít mimo jiné pro měření impulzních vlastností místnosti. Měření těchto vlastností slouží pro zjištění, zda je místnost vhodná například pro konference, přednášky, mítinky apod. Nahrávání pomáhá pomocí mikrofонů s 3D vytisknutými koncovkami, které se do systému zapojují XLR konektory.

- 1 - Mikrofony
- 2 - XLR konektory
- 3 - Napájecí kabel
- 4 - Ethernetový kabel

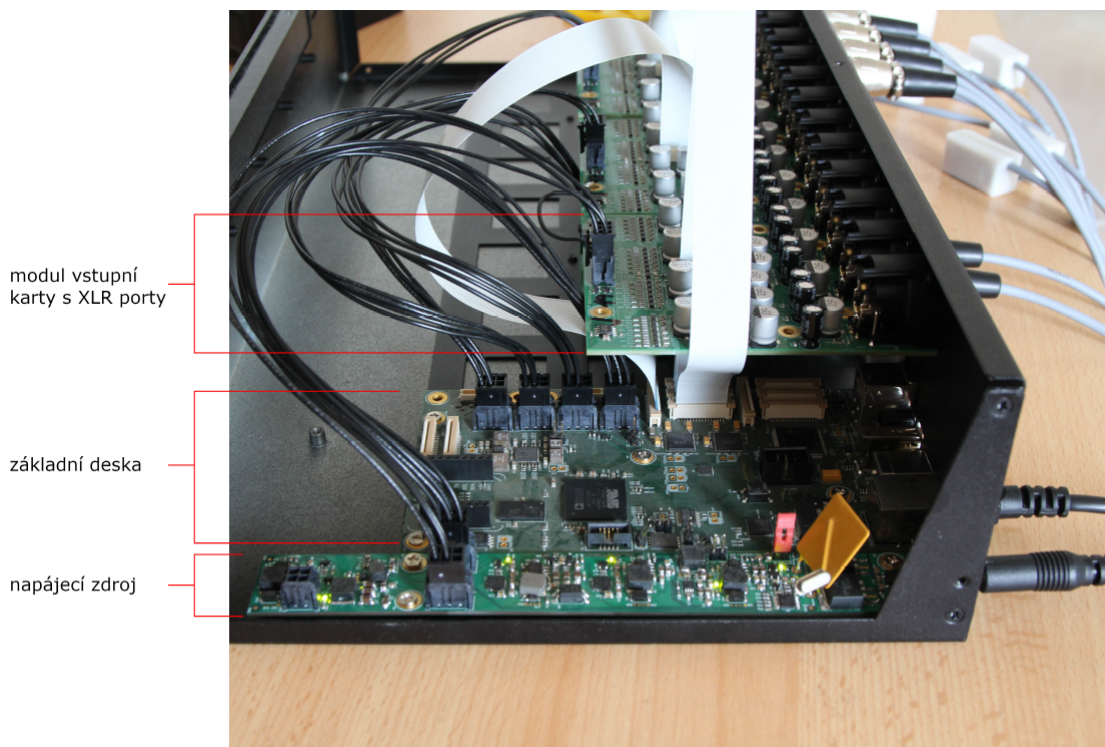


Obrázek 2.1: Schéma zapojení zvukového systému.

Uvnitř tohoto černého kvádru se skrývá zdroj napájení, základová deska a moduly zvukových karet. Na základní desce je čip ADSP-sc589 od firmy Analog Devices¹. Tento procesor je primárně určen ke zpracování audia. Obsahuje dvě SHARC jádra a ARM procesor Cortex A5. K čipu je připojena volatilní paměť, slot mikro SD karty pro bootování OS Linux a flash paměť pro trvalé uložení souborů (například program pro SHARC jádra). Lokální

¹analog.com/en

úložiště souborů (např. audio soubory pro přehrávání sinových signálů) je na zmíněné SD kartě uvnitř zařízení.



Obrázek 2.2: Interiér zvukového systému. Převzato z Diplomové práce pana Martina Lacha.

DSP SHARC

Digitální Signálový Procesor (DSP) SHARC slouží k provádění operací s plovoucí řádovou čárkou. Je přímo připojen k A/D převodníku a zpracovává všechna získaná audio data. K přístupu k datům tohoto procesoru slouží ALSA ovladač.

Cortex A5

Jedná se o 32 bitový ARM procesor z řady ARMv7-A, rozšířený o operace s plovoucí řádovou čárkou. Využívá L1 instrukční a datovou cache (každá 32 kB) a 256 kB L2 cache.

Na zadní straně systému se nachází vstup pro napájecí kabel, který je dodán se systémem, vstup pro Ethernetový kabel, sloužící pro připojení počítače a jeho následné ovládání a XLR konektory pro připojení mikrofónů nebo reproduktorů. Dodaná konfigurace systému obsahuje čtyři zvukové karty, každá se čtyřmi vstupy pro mikrofony, což činí šestnáct konektorů pro mikrofony. Tyto konektory se nachází v horní řadě, v řadě spodní jsou výstupy pro připojení reproduktorů, chceme-li současně s nahráváním i přehrávat nějaké zvukové stopy (typicky bílý/růžový šum, sinové signály apod.)

Zvukový systém nelze ovládat přímo, má na sobě pouze jedno tlačítko, a to kolébkový spínač pro zapnutí/vypnutí přístroje. Je nutné jej propojit ethernetovým kabelem s počítačem a ovládat z něho. Po spuštění přístroje se rozsvítí LED dioda v blízkosti spínače a začne načítání systému.

2.2 Boot

Systém je načten z SD karty do paměti. Při každém vypnutí systému dojde ke smazání volatilní paměti a při dalším spuštění se systém bootuje znovu. Veškerá data jsou ztracena. Pokud je potřeba data uložit, je nutno je ukládat na externí úložiště či na danou SD kartu, na jiný oddíl, než který obsahuje operační systém Linux.

Jakmile je systém naboťovaný, nastaví síť, spustí SHARC jádra a nakonec spustí script, jímž nastaví hodnoty *GAIN* a *PHANTOM* jednotlivých mikrofonů a spustí aplikaci nazývanou Streamer (v paměti systému je binární soubor aplikace streamer pojmenován *streaming-app-16ch*).

2.3 Streamer

Aplikace *streaming-app-16ch* (dále jen Streamer) spuštěná po naboťování systému čeká na připojení programu Receiver, komunikuje s ním, čte audio data z SHARC jádra a odesílá je Receiveru.

Streamer je spuštěn následujícím příkazem:

```
streaming-app-16ch 0.0.0.0 80 /srv/www ,
```

kde:

- **0.0.0.0** - lokální adresa pro spuštění serveru
- **80** - port, na kterém probíhá komunikace (80 - **http**)
- **/srv/www** - lokace webové aplikace pro ovládání systému.

Main

Soubor *Main* je jedním ze zdrojových souborů programu a také jeho počátkem při spuštění. Je zde provedena kontrola argumentů spuštění, při úspěchu je vytvořen a inicializován objekt třídy *CStreamerController*. Poté je vytvořen server pro webovou aplikaci pro ovládání, *CStreamerController* je na něho napojen a čeká na instrukce.

CStreamerController

Třída *CStreamerController* slouží k vytvoření a inicializaci objektu **streamer** třídy *CAlsaStreamer*, vyřizuje řídicí zprávy a implementuje rozhraní pro přístup ke Streameru.

Při jeho tvorbě *StreamerController* vytvoří objekt *CNetworkConnetion* sloužící pro navázání spojení mezi Streamerem a Receiverem. V následující inicializaci *StreamerController* vytvoří objekt třídy *CAlsaStreamer* spolu se spojením s Receiverem, a na tomto spojení spustí Streamer naslouchání pro příjem řídicích zpráv z Receiveru.

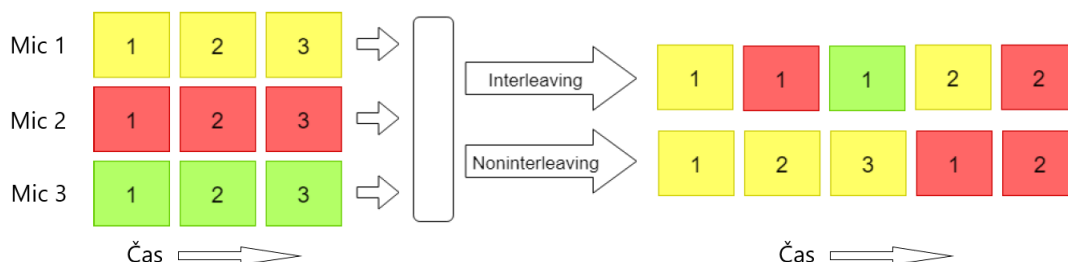
StreamerController přímo ovládá Streamer - používá jeho metody k řízení běhu aplikace. Pokud například přijde z Receiveru žádost o data, *StreamerController* zprávu zpracuje a zavolá příslušné metody objektu **streamer**.

CAlsaStreamer

CAlsaStreamer rozšiřuje jednoduchou třídu *IStreamer*. Poté, co je vytvořen objekt **streamer** třídy *CAlsaStreamer*, jsou jeho atributy naplněny počátečními hodnotami, inicializuje svůj kruhový buffer (*CCircBuffer*) a inicializuje zvukovou kartu.

Okamžitě po inicializaci je zahájeno čtení vzorků ze zvukové karty a jejich následná analýza. Při streamování se všechna data posílají do sítě. Proto je mezi čtením a odesláním kruhový buffer, který uloží několik vzorků před odesláním a tím vyrovnává případnou prodlevu vzniklou při postupném odesílání paketů.

Data získáváme v prokládaném formátu (Obrázek 2.3). Není vhodné data rozdělovat, ideální je zarovnání na násobek počtu využívaných kanálů. Tímto způsobem je též zaručena časová synchronizace vzorků. Odesílání dat probíhá ve chvíli, kdy o ně Receiver požádá.



Obrázek 2.3: Prokládání vzorků. Převzato z diplomové práce pana Martina Lacha[4], upraveno.

2.4 Receiver

Pro komunikaci a ovládání zvukového systému byl vytvořen program *Receiver*. Tento program je nutné spustit na připojeném počítači:

```
./Receiver --IP <IP adresa zvukového systému>
```

Dalšími možnými argumenty při spuštění Receiveru jsou:

- **--out-dir** </cesta/do/složky> - umístění, kam se budou ukládat nahraná data. Výchozí umístění je ./, tudíž se získané soubory uloží přímo do složky, kde se nachází Receiver.
- **--out-file** <název> - zadání prefixu pro pojmenování vytvořených souborů. Název souboru je potom <název> následovaný číslem kanálu, ze kterého data přišla (např. *out_ch01* - *out_ch16*).
- **--log-file** <název> - určení názvu souboru se záznamy z průběhu nahrávání.
- **--show-level** - zobrazení informací o probíhající nahrávce po určitých časových intervalech.

Pro ukládání dat je zde opět využit kruhový buffer. Kvůli režii je výhodnější uložit do souboru až větší množství dat a zároveň se zde třídí prokládané vzorky z jednotlivých kanálů. Pro každý kanál je zde jeden kruhový buffer.

2.5 Komunikace mezi Streamerem a Receiverem

Tyto dva programy spolu komunikují pomocí protokolu pro síťovou komunikaci na portu 80. Vyměňují si a zpracovávají různé pakety. Tyto pakety lze rozdělit na:

- datové pakety,
- konfigurační pakety a
- informační pakety.

Struktury jednotlivých paketů lze vidět na Obrázku 2.4.

Datové pakety

Slouží ke streamování dat. Paket se skládá z typu paketu, časové známky pro určení pořadí přijatých paketů, informace o počtu vzorků a samostatných audio dat.

Konfigurační pakety

Tyto pakety jsou využity pro ovládání jednotlivých programů, například ke spuštění nahrávání. Obsahují informace o typu paketu, kód specifikující zprávu v paketu a hodnotu, která nabývá významu v závislosti na typu zprávy.

Informační pakety

Informační pakety reprezentují řídicí či jiné informace. Jsou tvořeny typem paketu, kódem rozlišujícím jednotlivé typy informací a částí „název souboru“, která nejčastěji obsahuje názvy souborů, které jsou programy vyžádány (např. při přehrávání zvukových stop).

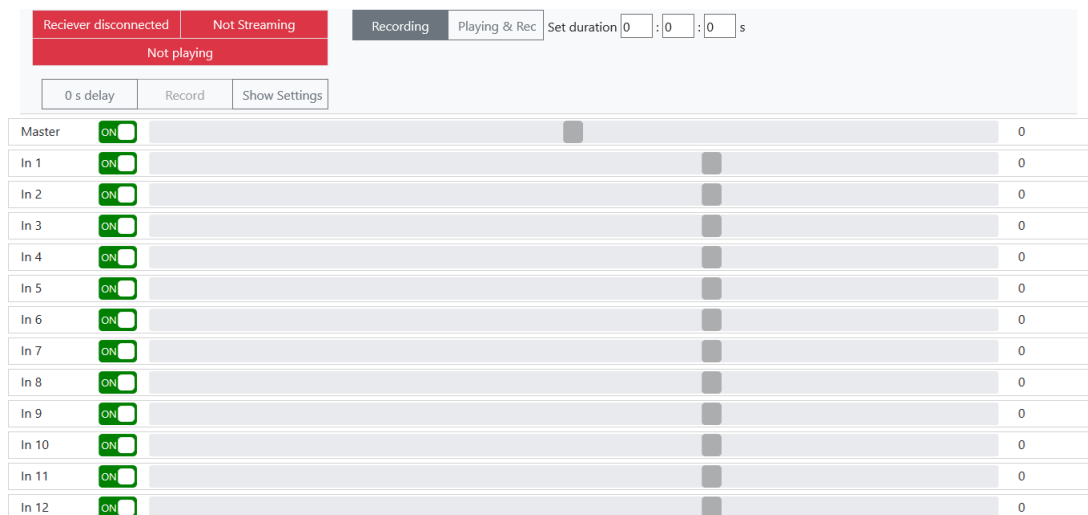
	0B	1B				5B				9B	PACKET_SIZE
Typ paketu											
Datový paket	typ	časová známka				počet vzorků				data	→
Konfigurační paket	typ	kód				hodnota				nedefinováno	
Informační paket	typ	kód				název souboru				→	FILENAME SIZE

Obrázek 2.4: Typy paketů s využitím paměti. Převzato z Diplomové práce pana Martina Lacha[4]

2.6 Vizualizační a řídicí internetová aplikace

Jde o webovou aplikaci přístupnou pouze lokálně (náhled lze vidět na Obrázku 2.5). Po zapnutí systému (server se spouští hned po bootu) a připojení počítače lze zadat IP adresu do prohlížeče. Aplikaci jsou po požádání předány potřebné soubory, styly a JavaScript pro generování progress barů jednotlivých kanálů. Není zapotřebí připojení k internetu.

Pro plnou funkčnost webové aplikace je třeba spustit program *Receiver*. Po spuštění Receiveru zezelená položka **Receiver disconnected** a text se změní na **Receiver connected**. V tomto okamžiku je aplikace připravena k použití. Zpřístupnilo se tlačítko **Record**, které slouží k aktivaci streamování dat ze Streameru do Receiveru. Ten tato data zpracuje a uloží.



Obrázek 2.5: Ukázka webové aplikace

Je zde možnost volby mezi pouhým nahráváním a přehráváním s nahráváním. Volba probíhá změnou zvýrazněné položky **Recording** nebo **Playing & Rec.**. Druhá ze zmíněných slouží v případě, že máme k zvukovému systému připojen i reproduktor. Dle volby v aplikaci je z paměti systému přehrávána jedna nebo více zvukových stop. Často se jedná se o tzv. růžový šum, bílý šum a sinový signál, které slouží k měření akustických vlastností místnosti.

Server je tvořen `http` rozhraním, které volá funkce dle potřeby vlastní či uživatele. Data od uživatele jsou přijata v `GET` požadavku. Odpovědi serveru jsou ve formátu `JSON` pro snadné zpracování JavaScriptem.

Pro tvorbu webové aplikace byla využita rozsáhlá `C++` knihovna Boost². Řadí se mezi standardní knihovny a je využívána firmou Audified. Knihovna je již součástí operačního systému.

²<https://www.boost.org/>

Kapitola 3

Analýza problému

Zvukový systém je plně funkční a může být využit v praxi k nahrávání i k měření impulzních vlastností místnosti. Jeho první spuštění však může být náročné. Dokumentace k systému prakticky neexistuje, je nutné čerpat z komentářů zdrojového kódu. V kapitole 4 je k nalezení návod pro první spuštění a operace se systémem současně s úpravami pro lepší přívětivost systému.

Software systému potřebuje různé úpravy a vylepšení, ať už se jedná o vizualizaci získaných dat ve webové aplikaci nebo přijímání argumentů spuštění jednotlivých komponent.

3.1 Streamer a Receiver

Součinnost těchto programů a jejich vlastní průběh není jednoduchý. Zdrojové kódy jsou většinou stroze okomentované, činnost programu je tedy nutné zjistit metodou *reverse engineering*.

Nastavování gainů (upravení zisku předzesilovače pro zvýšení/snížení míry vybuzení) probíhá pomocí souborů `/proc/asound/SHARC/gain#[0-15]`, pro každý kanál jeden. Gain se udává v decibelech, hodnoty jsou 0-6 (0 = 0dB, ..., 6 = 60dB). Výchozí hodnota nastavení gainu je 4. Ovládání gainů z webové aplikace přepisuje zmíněné soubory na nové hodnoty. Je třeba spustit nějaké audio (třeba sinový puls), nastavit gainy, zkontrolovat, případně proces opakovat. Tato práce však může být zdlouhavá, proto by bylo vhodné vytvořit automatické nastavení gainů. Z externího či připojeného reproduktoru bude vysílán zvuk, podle kterého si program sám nastaví vhodné hodnoty gainů, aby byly zachyceny potřebné frekvence.

Receiver pro uložení získaných audio souborů vytvoří nový adresář, kam tyto soubory uloží. Název může uživatel specifikovat, avšak chybí podpora maker pro zadání data a času.

Aplikaci Streamer je potřeba zkompilovat na systému, kde má běžet. Jelikož vestavěné systémy mají omezené možnosti, využíváme křížovou kompilaci, kdy hostitelský systém dokáže zkompilovat kód pro chtěný systém. V současnosti je počet využívaných kanálů definován v makru `NUM_CHANNELS` v souboru `AudioStreamer_definitions.h` a pokud chceme systém využívat s 32 kanály, je potřeba toto makro ve zdrojovém kódu přepsat a znovu křížově kompilovat. Toto je velmi neefektivní způsob, je potřeba jej vyřešit.

V kódu je poměrně častý výskyt překlepů a komentářů. Jedním z markantních omylů je záměna písmen ve slově receiver, který je na mnoha místech pojmenován *reciever* či *receiver*. Chyba je důležitá například při spouštění programu Receiver, který je vlastně pojmenován Reciever.

3.2 Webová aplikace

Aplikace dobře plní svůj účel. Komunikace s Receiverem je bez problému a změny nastavení jsou provedeny. Je jednoduše pochopitelná a přímočará. Díky této aplikaci lze zvukový systém ovládat pohodlně bez nutnosti psaní příkazů do konzole (samozřejmě kromě spuštění programu Receiver). Je zde i možnost vytvoření playlistu pro nahrávání s připojeným reproduktorem pro přehrávání šumů.

Jedním z problémů aplikace je fakt, že i pokud je připojeno méně než 16 mikrofونů, po nahrávání je vytvořen soubor s audio daty pro každý z 16 kanálů. Tato skutečnost je nepříjemná, protože pro dva mikrofony nepotřebujeme dalších čtrnáct „prázdných souborů“. Současně s těmito 16 soubory jsou vytvořeny dva derivační soubory pro analýzu nahrávání a záznam o průběhu nahrávání.

Po zapnutí nahrávání jsou data vizuálně interpretována na obrazovce. Pro každý kanál je zde progress bar, jehož zaplnění udává míru vybuzení daného mikrofону. Posuvníkem lze specifikovat různé hodnoty GAIN. Aktualizace zobrazování vybuzení je však z důvodu nepřetěžování systému nastavena na 1s. Důvodem je nedostatečný výkon komponent zařízení, při vyšší frekvenci aktualizace informací dochází k výpadkům při sběru vzorků z mikrofонů.

Vzhled stránky si také žádá úpravu. Některé součásti jsou prakticky bez stylu a celkový vzhled vypadá, jako by plýtlval místem. Současně by bylo vhodné implementovat omezení ukládání dat z jednotlivých kanálů. Není vždy potřeba všech 18 souborů.

3.3 Metadata Wizard

Součástí procesu měření impulzní odezvy je sbírání informací o prostředí, ve kterém se systém nachází. Díky tomu mohou být vzorky z různých místností lépe porovnatelné. Tato metadata jsou nejčastěji uložena v souboru formátu `.csv`.

Jde o data jako ID mikrofону, rozměry místnosti, materiál stěn, umístění mikrofонů, jednotlivé nastavení gainů mikrofонů a obdobná metadata.

Proces zaznamenávání metadat je v současné době velmi monotónní a zdoluhavý, je proto nutné jej usnadnit a zpříjemnit. Nyní jsou metadata zanášena ručně do souborů `.csv` nebo do tabulek v podobném formátu (typ - hodnota). Příklad takové tabulky je k nalezení na Obrázku 3.1. Tento způsob je náchylný k překlepovým chybám, které jsou kvůli špatné orientaci špatně dohledatelné. Tento způsob by mohl být zjednodušen vytvořením webové aplikace, jakéhosi průvodce zaznamenáváním metadat, která uživatele provede celým procesem a udá jasný postup kdy a jak které údaje zapsat. Aplikace by měla také obsahovat kontroly vstupů, aby uživatel věděl, zda při zapisování metadat neudělal chybu či nedošlo k odchyldám měření.

Vytvoření takovýchto metadat je časově náročná monotónní práce. Pro usnadnění zaznamenání údajů o místnosti a zároveň jako návod pro postup měření bude sloužit webová aplikace provádějící uživatele celým procesem obsluhy přístroje. Pro implementaci byl zvolen jazyk PHP, konkrétněji framework Nette[3].

Nette

Nette je rodina samostatně použitelných komponent pro jazyk PHP 8. Tento framework se dle recenzí pyšní svou efektivní databázovou vrstvou, jednoduchostí a spolehlivostí. Byl vytvořen českými vývojáři a využívají jej významné společnosti, jako třeba GE Money, Mladá

Environment	
Variable	Value
\\$EnvID	3
\\$EnvName	VUT_FIT_D105
\\$EnvTemp	21
\\$EnvDescription	Large lecture room triangular shape for 300 students
\\$EnvPhoto1	
\\$EnvPhoto2	
\\$EnvPhoto3	
\\$EnvPhoto4	
\\$EnvPhoto5	
\\$EnvPhoto6	
\\$EnvDepth	17.2
\\$EnvWidth	22.791
\\$EnvHeight	6.9
\\$Env2Depth	
\\$Env2Width	
\\$Env2Height	3.38
\\$EnvVolume	
\\$EnvType	Hall
\\$EnvSubType	Large
\\$EnvBCKNoiseLevel	34
\\$EnvMatWall	concrete
\\$EnvMatFloor	wood
\\$EnvMatCeiling	Plasterboard
\\$EnvFurniture	30

Obrázek 3.1: Příklad metadat v programu MS Excel

fronta, Slevomat, ESET a další. Pro vytvoření stránek je frameworkem Nette podporován šablonovací jazyk Latte[2].

Nette je MVP framework. Stojí na komponentách tří typů, každá z nich má jednu ze základních úloh - řízení, logika a výstup.

- **Model** - obsahuje aplikační logiku (výpočty, práce s databází). Většinou každá datová entita má svůj vlastní model (uživatel, příspěvek, komentář...).
- **Pohled** (V Nette Template, šablona) - šablony s HTML kódem vytvořeny šablonovacím jazykem Latte, který umožňuje do šablon vkládat data z PHP pomocí speciálních značek.
- **Presenter** - komponenta, se kterou komunikuje uživatel. Po předání parametrů je vrácena HTML stránka. Presenter předá parametry modelům, od kterých získá zpracovaná data a tato data předá šablonám, které je začlení do HTML kódu. Takto připravený HTML kód je zobrazen uživateli v prohlížeči.

Bylo by vhodné, aby tato aplikace byla vložena přímo do zvukového systému, jak tomu je u vizualizační a řídicí aplikace. Je však velmi pravděpodobné, že zavedení aplikace Webwizard na systém nebude možné z důvodu omezeného výkonu soustavy.

Kapitola 4

Návod k obsluze

Jak již bylo zmíněno, ke zvukovému systému není prakticky žádná dokumentace kromě komentářů zdrojových kódů, příručky pro instalaci, upgrade a správu operačního systému a diplomové práce pana Martina Lacha[4]. Neobsahuje však rady jak zprovoznit systém a provádět práci, kterou potřebujeme.

4.1 Připojení

Zapojení není nijak složité. Správně zapojený přístroj lze vidět na Obrázku 2.1. Pokud je vše správně zapojeno, systém se stisknutím spínače zapne a v momentě, kdy je připraven, svítí modrá LED dioda u spínače. Pokud svítí červená, systém se stále načítá.

Na propojení ethernet kabelem s počítačem časově nezáleží, lze tak učinit před, při i po spuštění systému. Pro správnou komunikaci je však potřeba změnit nastavení síťového adaptéru. V rozšířených možnostech nastavení adaptéru je nutné zakázat automatické nastavení IP adresy a manuálně zadat IP adresu ve stejné podsíti, jako je zařízení. Aktuální IP adresa zařízení je **147.229.14.27**. Pro laiky - je třeba změnit poslední oktet (trojčíslí; zde 27) na jiné, nejlépe v rozmezí 1 - 99.

4.2 Křížová kompilace

Křížová kompilace slouží ke kompilaci programu pro prostředí jiného systému při nemožnosti kompilovat přímo v daném systému. Zde je to očekávané a vhodné řešení, jelikož vestavěný systém trpí nižším výkonem a nepřítomností kompilátoru. Dobrou volbou je využití toolchainu pro generování vestavěného Linuxu Buildroot¹. Nástroj byl nakonfigurován firmou Audified pro daný operační systém.

Tento toolchain je připraven pro operační systém **Ubuntu 18.04.1**, ale měl by fungovat i na mírně odlišných distribucích. Je nutné toolchain stáhnout a rozbalit na zařízení s daným OS a ve stejném adresáři spustit přiložený skript **relocate-sdk.sh**.

V tomto momentě je možné spustit **Makefile** pro Streamer a program je úspěšně zkompileován do jediného binárního souboru. Tento soubor je možné přenést na zařízení a z něho jej spustit.

¹<https://buildroot.org/>

4.3 Streamer

Nyní je možné se programem `ssh` (v případě Windows je výbornou možností aplikace Putty²) připojit na systém. V něm je vytvořen uživatel **adsp** bez práv administrátora. Připojení do systému tedy může vypadat následovně:

```
ssh adsp@147.229.14.27
```

Pro přihlášení je třeba zadat heslo, které je stejné jako název uživatele - **adsp**. Po autentizaci můžeme procházet adresářovou strukturu zařízení a volat určité programy. Přístup je však omezen, nemůžeme například spustit a otestovat vlastní verzi Streameru, jelikož již na dané IP adrese a portu Streamer běží; byl spuštěn při bootování systému. Pro vyšší práva je třeba změna uživatele na root (příkaz `su`). V tomto momentě je možné zastavit proces příkazem `kill` a spustit vlastní program.

```
streamer 0.0.0.0 80 /srv/www 16
```

kde:

- **0.0.0.0** - lokální adresa pro spuštění serveru
- **80** - port, na kterém probíhá komunikace (80 - http)
- **/srv/www** - lokace webové aplikace pro ovládání systému.

Přidané číslo 16 na konci příkazu udává počet kanálů. Pokud chceme využít jiný server než ten v umístění `/srv/www`, musíme ho do systému nahrát a zadat cestu k němu.

Důležitým omezením v připojení do systému je odmítnutí přístupu při pokusu o úpravu souboru jinde, než ve složce `/home/adsp/`. Jde o bezpečnostní opatření pro zabránění nechtěné změny některého ze souborů. Všechny soubory, které je potřeba přenést do systému, je nutné umístit do dané složky `/home/adsp/`.

Program Streamer dříve pracoval s pevně definovaným počtem kanálů pomocí makra. Při přechodu na 32 kanálů bylo třeba makro změnit a znovu křížově kompilovat. Nyní je možné pouze zastavit proces Streameru a spustit ho znovu s přidaným argumentem 32 (`./streamer 0.0.0.0 80 /srv/www 32`).

Jestliže je ve zvukovém systému trvale nahrána a spouštěna správná verze Streameru, není potřeba se na systém připojovat příkazem `ssh`, ovládání je možné z programu Receiver.

4.4 Receiver

Receiver je vlastně ovladač Streameru. Komunikuje s ním na HTTP portu 80. Nejjednodušší příklad spuštění Receiveru je následující:

```
./Receiver --IP 147.229.14.27.
```

Takto je Receiver spuštěn s výchozím nastavením a vyčkává na vstup uživatele. Výchozí nastavení mimo jiné zahrnuje „tichý režim“, kdy je do konzole vypisováno jen minimum výstupů. Pro zobrazení většího množství informací slouží argument `--show-level`.

²<https://www.putty.org/>

Program je možné spustit s přídatnými argumenty:

- **--out-dir** *</cesta/do/slozky>* - umístění, kam se budou ukládat nahraná data. Výchozí umístění je *./*, tudíž se získané soubory uloží přímo do složky, kde se nachází Receiver.
- **--out-file** *<název>* - zadání prefixu pro pojmenování vytvořených souborů. Název souboru je potom *<název>* následovaný číslem kanálu, ze kterého data přišla (např. *out_ch01* - *out_ch16*).
- **--log-file** *<název>* - určení názvu souboru se záznamy z průběhu nahrávání.
- **--show-level** - zobrazení informací o probíhajícím nahrávání po určitých časových intervalech.

Tyto argumenty jsou také zobrazeny po spuštění programu s argumentem **--help** nebo **-h**.

Změnou oproti původnímu návrhu prošlo zpracování parametru **--out-dir**. Pokud není zadán, defaultně Receiver vytvoří složku ve stejném adresáři, jako se sám nachází, pojmenuje ji *Recording_<%C>channels_<%D>-<%T>*, kde *%C* je počet nahrávaných kanálů, *%D* je datum formátu ROK-MĚSÍC-DEN a *%T* značí čas formátu HODINY:MINUTY:VTEŘINY. Při nutnosti zadat jiný adresář lze tyto zkratky využít, například **--out-dir** *%C_%D_%T* vytvoří složku se jménem *16_2001-01-01_00.00.00* (dle aktuálního nastavení systému). Do této složky se budou ukládat soubory s nahranými daty. Název ukládaných souborů lze samozřejmě změnit argumentem **--out-file**.

Spuštěním Receiveru a připojením na systém jej můžeme ovládat přímo z konzolové aplikace. Po stisknutí některé z následujících kláves a poté stlačením klávesy Enter dojde k vykonání příslušné instrukce:

- **s** (start) - spustí ukládání audio dat z programu Streamer,
- **e** (end) - pozastaví ukládání dat; v tomto momentě je možné opět ukládání spustit příkazem **s**, další data jsou ukládána za již stávající v existujících souborech - nedojde k mezeře mezi daty, ale úsek, kdy bylo ukládání pozastaveno, uložen není,
- **q** (quit) - zastaví nahrávání a ukončí program Receiver.

Dříve bylo možné ovládat Receiver pouze příkazy **s** a **e**, které spouštěly a ukončovaly program. Průběh byl takový, že po příkazu **e** byl program Receiver ukončen bez možnosti opětovného spuštění nahrávání a bez oznámení Streameru, že byl Receiver zastaven. To Streamer zjistí až po neúspěšném udržení spojení. Za tímto účelem byla implementována možnost **q** a změněn význam **e** pro lepší práci se systémem.

Z Receiveru nelze ovládat přehrávání audio souboru ze zvukového systému. To je možné pouze ve webové aplikaci zvolením možnosti *Play & Record* a následným vybráním souborů pro přehrání. Délka času nahrávání závisí pouze na uživateli a dostatečném množství volné paměti pro ukládání dat. Současně nastavený systém byl testován tak, že bylo spuštěno ukládání po dobu 48 hodin. Bylo zjištěno, že systém dokáže beze ztrát ukládat data po celou dobu běhu a tímto způsobem bylo vytvořeno 560 GB dat v osmnácti souborech.

4.5 Webová aplikace

Pro chod webové aplikace je zapotřebí správně spuštěný Receiver připojený na Streamer ve zvukovém systému. Pokud je v levém horním rohu zeleně podbarvené pole **Receiver connected**, je možné systém ovládat. Pod těmito ukazateli se nachází tlačítka pro ovládání nahrávání. Nejdůležitějším je tlačítko **Record** (v případě nahrávání se současným přehráváním zvuku **Play & Record**), kterým se spustí a následovně ukončí nahrávání. Funkce je stejná, jako příkazy **s** a **q**. V praxi to znamená, že po stisknutí tlačítka je zahájeno ukládání audio dat a po opětovném stisknutí dojde k ukončení programu Receiver.

V aplikaci lze nastavit trvání nahrávání vyplněním políček **Set duration**. Časovač funguje správně, avšak po vypršení limitu je opět běh Receiveru zastaven jako při příkazu **q**.

4.6 Proces zaznamenávání metadat

Zaznamenávání metadat slouží k možnosti porovnání výsledků různých měření v závislosti na vlastnostech prostředí a použitých zařízení (mikrofonů, reproduktorů). Data jsou uložena ve formátu **csv**, tedy **TYP, HODNOTA**.

Postup při zaznamenávání metadat o nahrávání je prostý. Uživatel:

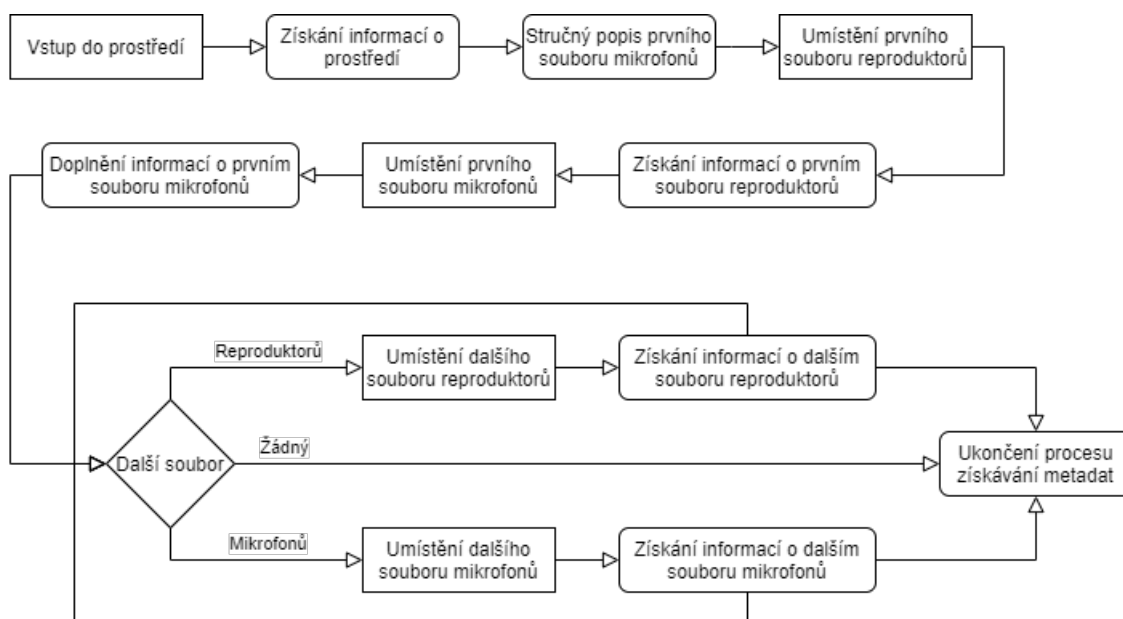
1. vstoupí do prostředí, které chce měřit,
2. zapojí a zprovozní nahrávací zařízení a počítač, kam se vše bude ukládat,
3. zapíše údaje o prostředí
4. rozmístí reproduktory,
5. zapíše údaje o reproduktorech,
6. rozmístí mikrofony,
7. zapíše údaje o mikrofonech,
8. provede nahrávání audia,
9. změni pozice reproduktorů/mikrofonů a opakuje body 4 - 8,
10. ukončí proces měření akustických vlastností prostředí.

Pro lepší přehled byl vytvořen graf zobrazující tento postup (Obrázek 4.1).

4.7 Aplikace Webwizard

Webová aplikace Webwizard slouží k usnadnění zaznamenávání metadat o nahrávání. Aplikace uživateli šetří čas při jejich zapisování a obsahuje rady, když něčemu nerozumí. Po instalaci této lokální webové aplikace a jejím zprovoznění na některém webovém serveru je aplikace připravena k používání.

Po spuštění prohlížeče lze zadat odkaz `127.0.0.1/webwizard/www`, nebo významově stejný `localhost/webwizard/www`, což způsobí načtení Domovské stránky této aplikace.



Obrázek 4.1: Graf procesu zaznamenávání metadat

Pokud již uživatel s aplikací pracoval a vytvořil jeden či více souborů s metadaty, jsou tyto soubory zobrazeny v seznamu v pravé části Domovské stránky. Pokud chce uživatel definovat název souboru s metadaty, může tak učinit právě zde. Ve většině případů však toto pole nemusí být využito, jelikož bez jeho vyplnění je jako název souboru použit unikátní název prostředí (např. *Kuchyň_u_mě_doma*, poté bude název vytvořeného souboru *Kuchyň_u_mě_doma.csv*), který bude zadán při průchodu aplikací. Pro zahájení procesu zapisování metadat slouží tlačítko „Submit and Begin“.

Aplikace provede uživatele procesem nahrávání audio dat v místnosti. Nezná-li uživatel postup, jakým tato data získat, poradí formulářů v aplikaci mu tento postup jasně určí. Současně je postup zobrazen na Obrázku 4.1. Rámečky s hranatými okraji popisují akce uživatele, kulaté rámečky značí práci s webovou aplikací. Každý kulatý rámeček symbolizuje jeden krok ve webové aplikaci.

Při definici setu reproduktorů lze do jednoho setu přidat až deset reproduktorů, v případě mikrofonů lze specifikovat až třicet dva mikrofonů.

Na každé stránce je v horní části navigační lišta, díky které můžeme přejít zpět na Domovskou stránku (pozor - pokud se rozhodneme přejít na Domovskou stránku, budou veškerá vyplněná data ztracena), nebo zobrazit okno s radami pro danou stránku.

Současně všechny hodnoty, které lze ve formulářích vyplnit, jsou opatřeny stručným a výstižným popisem, co znamenají, kdyby uživatel tápal.

Průchod aplikací končí na stránce Finish, kde je možné editovat vyplněné formuláře, najít informace o chybách při vyplňování a stáhnout požadovaný soubor, obsahující zaznamenaná data. Tento soubor je vytvořen ve formátu *csv*, tedy každý řádek reprezentuje jeden údaj ve formátu **název, hodnota**. Soubor je možné otevřít v textovém editoru či tabulkovém procesoru (MS Excel, LibreOffice a.j.).

Kapitola 5

Webová aplikace Webwizard

Pro usnadnění zaznamenávání potřebných metadat byla vytvořena webová aplikace Webwizard. Tato aplikace byla vyvinuta, aby uživateli celý proces získávání a zapisování metadat co možná nejvíce usnadnila.

Jde o webovou aplikaci zpřístupněnou pouze lokálně. Pro účely získávání metadat je zbytečné, aby byla aplikace veřejně přístupná na internetu, uživatel při měření akustických vlastností místnosti nepotřebuje přístup k internetu. V mnoha případech ani připojení nemusí být dostupné.

Pro instalaci aplikace na nové zařízení však přístup k internetu zapotřebí je, jelikož je nutné nainstalovat HTTP server (Apache2 či Nginx) a jazyk PHP 8.0.

5.1 Použité technologie

Pro tvorbu aplikace bylo využito různých technologií určených k tvorbě webového serveru a webových stránek. Aplikace byla vyvinuta na operačním systému Linux, konkrétně **Ubuntu 18.04**, jelikož je pro programátora velmi pohodlný a také je tento operační systém využíván pro práci se zvukovým systémem, který vytváří nahrávky prostředí.

PHP 8.0[1]

Aplikace Webwizard byla implementována v jazyce PHP verze 8.0. PHP (rekurzivní zkratka PHP: Hypertext Preprocessor) je programovací jazyk určený především pro programování dynamických internetových stránek a webových aplikací. Zde byla jazykem PHP implementována serverová část aplikace zajišťující zpracování vstupů, vytváření dočasných souborů a jejich ukládání k pozdějšímu stažení na uživatelský počítač.

Framework Nette[3]

Tento framework implementuje řadu komponent využívaných pro tvorbu webových stránek a aplikací. Verze využitého frameworku Nette je 3.1. Nejvyužívanějšími komponentami v této aplikaci jsou formuláře a jejich subkomponenty - textová pole, selectboxy, tlačítka... . Nette 3.1 využívá PHP 7.4, nebo novější. Framework Nette obsahuje podporu pro šablonovací jazyk Latte, který usnadňuje tvorbu vizuální části aplikací. Díky Latte je v HTML kódu možné využít proměnné ze serverové části aplikace - není nutné je získávat AJAXovými požadavky. Současně je díky Latte možné provádět operace jazyka PHP - vytváření a zapisování do proměnných, aritmetické či logické operace a mnoho dalších.

JavaScript

Je využíváný pro kontroly vstupů a další funkcionalitu. Prakticky všechna tlačítka zavolají nějakou JavaScriptovou funkci, která provede příslušné operace, od kontroly, zda zadané souřadnice neukazují mimo místnost, po zobrazování a schovávání oken s nápovědou.

jQuery

Tato knihovna je připojena z důvodu zjednodušení vybírání konkrétních elementů na stránce a následnou práci s nimi. Zajišťuje ale i možnost stisknutí klávesy Enter pro přesun ukazatele z jednoho pole formuláře do pole pod ním.

Apache2

Jedná se o často využívaný HTTP server umožňující běh serverových částí webových aplikací. Zde byl tento server využit díky předchozím zkušenostem s ním. Další možností byl server Nginx, který se díky nezkušenosti nepodařilo nakonfigurovat pro správný běh aplikace.

Tracy

Součástí projektu je i ladicí nástroj **Tracy** dodaný spolu s frameworkem Nette. Tento nástroj způsobí, že případné chyby se při vývoji či ladění aplikace neukážou jako klasické chybové stránky s `http` chybovými kódy. Místo toho je zobrazena stránka popisující nastalou chybu spolu se specifikací řádku a souboru, kde k ní došlo. Jestliže je nutné tento ladicí nástroj zapnout, stačí zakomentovat v souboru `webwizard/app/Bootstrap.php` řádek 17 (`$configurator->setDebugMode(false)`);

Composer

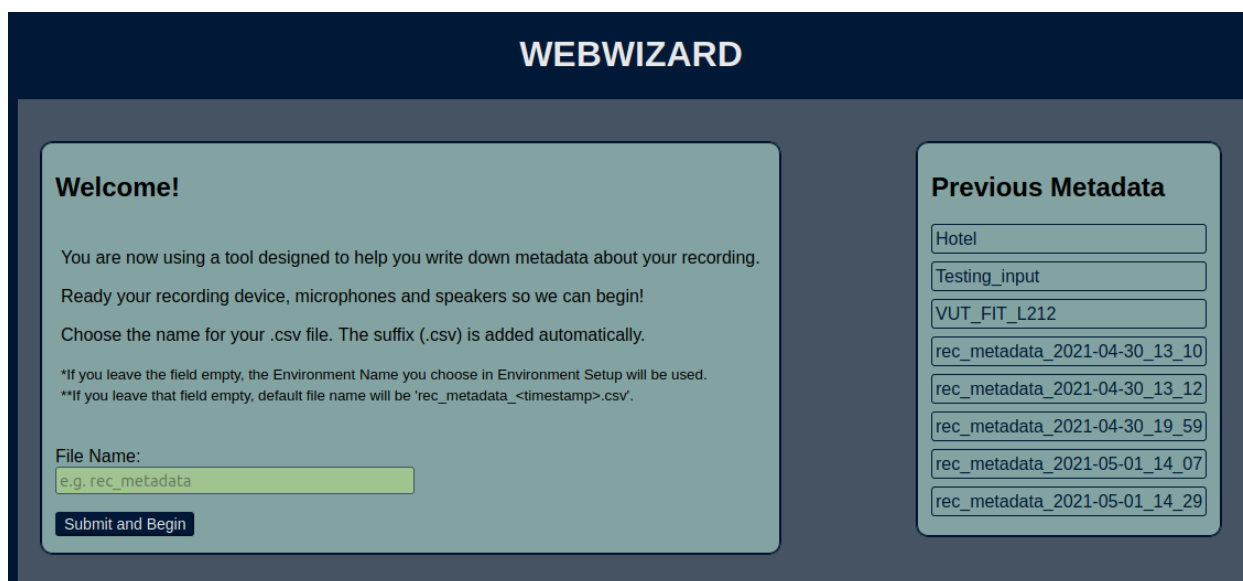
Composer (v překladu Skladatel) je nástroj pro správu závislostí projektu v PHP. Umožňuje jedním příkazem v terminálu přidat do projektu libovolnou knihovnu či již existující knihovny aktualizovat.

Současně lze tímto nástrojem založit projekt Nette, jak byla založena i aplikace WebWizard. Přidanou knihovnou oproti knihovnám již obsaženým v projektu Nette je pouze knihovna *contributte/forms-multiplier*, sloužící pro dynamické přidávání reproduktorů a mikrofónů do formuláře.

5.2 Používání a vzhled aplikace

Webwizard je zjednodušeně soubor formulářů pro zapisování metadat. Každému kroku při měření akustických vlastností místnosti odpovídá jedna stránka aplikace. Aplikace uživatele provede jednotlivými kroky získávání metadat a pomůže mu při některých nejasnostech. Jelikož je tato aplikace zpřístupněna pouze lokálně, je pro její používání po instalaci nutné zadat do prohlížeče adresu `127.0.0.1/webwizard/www`.

Vzhled aplikace (Obrázek 5.1) nebyl ničím inspirován. Jedná se o prostý soubor formulářů, které byly upraveny kaskádovými styly. Cílem byla jednoduchost zároveň s příjemným vzhledem. Pro vytvoření schématu barev, které jsou navzájem komplementární, byl využit



Obrázek 5.1: Ukázka domovské stránky aplikace Webwizard

nástroj *Coolors*¹. Tato stránka obsahuje náhodný generátor pětice barev, které spolu ladí. K těmto barvám lze přidat další podobné. Použity byly barvy spíše tmavší, aby pohled na stránku „nebil“ do očí.

Celá webová aplikace je výhradně v anglickém jazyce. Tato aplikace bude využívána na Fakultě informačních technologií VUT a její uživatel nemusí ovládat český jazyk.

5.2.1 Průchod aplikací

Po načtení adresy webové aplikace uživatele přivítá Domovská stránka, ze které lze jít dvěma směry: zobrazení předchozích metadat, která se nacházejí na daném počítači; nebo pokračovat do procesu zaznamenávání metadat o novém měření.

Proces zaznamenávání metadat začíná na stránce **Environment Setup**. Zde jsou vyplněny údaje o prostředí, kde probíhá nahrávání. Mimo jiné jsou zde pole pro rozměry prostředí, jeho typ a materiál, ze kterého jsou vyrobeny stěny, podlaha a strop. Může zde být specifikován i *Relativní bod*, od kterého jsou měřeny relativní souřadnice. Pokud není Relativní bod specifikován zde, jsou jako jeho souřadnice později využity souřadnice prvního reproduktoru z prvního souboru reproduktorů.

Následuje krátký formulář **MicInit Setup**, který slouží k tomu, aby uživatel rozmístil po prostředí mikrofony a zapsal název tohoto souboru mikrofونů spolu s jeho popisem. Tyto údaje jsou uloženy k prvnímu souboru mikrofونů, zbytek je doplněn později. Důvodem je Relativní bod. Ve většině případů je poloha mikrofону zadána relativními souřadnicemi. Tyto souřadnice nejsou měřeny od počátku souřadné soustavy, nýbrž od Relativního bodu. Proto nemohou být v tento moment relativní souřadnice mikrofونů zapsány, Relativní bod nemusí být zatím určen.

Po MicInit Setup přichází formulář **Speaker Setup**, ve kterém jsou definovány specifikace prvního souboru reproduktorů. Uživatel má v tuto chvíli rozmístěné mikrofony a může přistoupit k rozmístění reproduktorů. Pokud neurčil Relativní bod dříve, zvolí jeden z reproduktorů, který bude reprezentovat Relativní bod, a uvede jej jako „Speaker 1“.

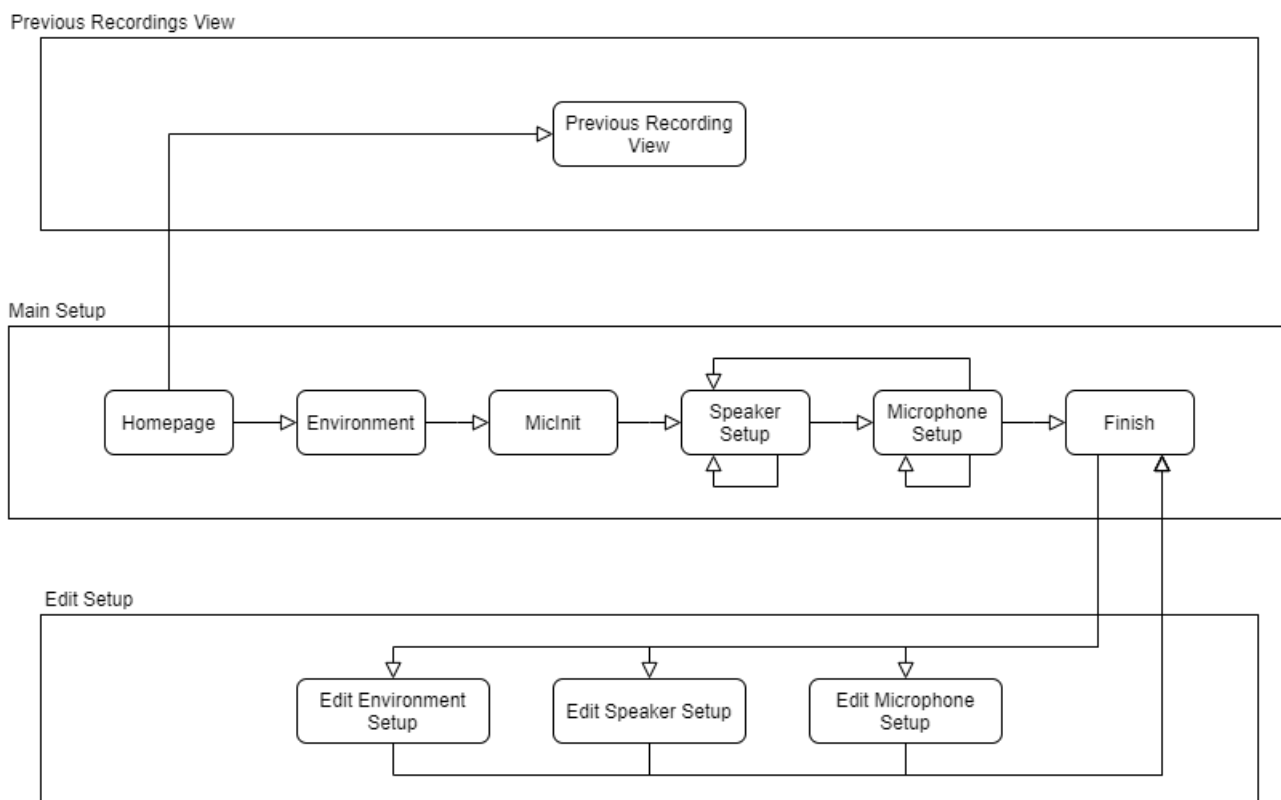
¹<https://coolors.co/>

Poté uživatele aplikace zavede na formulář **Microphone Setup**. Jedná se o doplnění mikrofونů do souboru mikrofونů, který byl vytvořen na stránce MicInit Setup. Je zde možné specifikovat umístění, typy a počet mikrofонů v daném souboru.

V tento moment má uživatel tři možnosti - přidat soubor reproduktorů, přidat soubor mikrofонů, nebo přejít na poslední část aplikace - **Finish Setup**.

5.2.2 Popis jednotlivých stránek

Každá stránka aplikace Webwizard obsahuje navigační lištu, na které je odkaz na Domovskou stránku, název aktuální stránky a tlačítko „HELP“, které uživateli napoví, co by měl dělat, nebo vysvětlí některé nejasnosti na stránce.



Obrázek 5.2: Graf toku práce s webovou aplikací Webwizard

Formuláře **Environment Setup**, **MicInit**, **Speaker Setup** a **Microphone Setup** jsou ovládány třídou **SetupPresenter**. Tato třída obsahuje metody pro vykreslení a funkcionality stránek. Díky frameworku Nette lze využít jeden *Presenter* pro více *Views*.

Homepage

Stránka vítá uživatele a nabízí dvě možnosti, co dál. Sekce vlevo uživateli umožní předem zadat název souboru s uloženými metadaty, který si uživatel později uloží na svůj počítač, společně s tlačítkem zahajujícím průchod aplikací. Vpravo se nachází odkazy na soubory, které byly na tomto počítači vytvořeny; tyto soubory jsou lokálně uloženy v adresáři `webwizard/tmp/rec/prev`.

Previous Setup

Stránka s pohledem na předchozí metadata je ovládána třídou **PreviousPresenter**. Obsahuje vždy data z jednoho souboru, který byl dříve vytvořen průchodem aplikace Webwizard. Zde jsou v tabulkách zobrazeny jednotlivé údaje k daným okruhům - **Environment**, **Speaker Setups** a **Microphone Setups**. Pro lepší přehlednost lze jednotlivé sekce sbalit či rozbalit tlačítkem **X** vpravo nahoře každé sekce. Tato stránka neslouží k editaci, pouze k zobrazení.

Environment Setup

Obsahuje dvě sekce - obecné informace o prostředí (identifikátor, jméno, naměřená teplota, popis, datum) a konkrétní informace (rozměry prostředí, objem, typ, podtyp, hladina ruchů v pozadí, materiály stěn, podlahy a stropu, procentuální obsazenost podlahy nábytkem). V konkrétních informacích je po stisknutí tlačítka *Set Relative Point Manually* možné zadat relativní bod, od kterého budou měřeny relativní souřadnice jednotlivých reproduktorů/mikrofonů. Pokud tento bod není zadán zde, je první reproduktor z prvního souboru reproduktorů určen jako relativní bod prostředí.

Pro pokračování je nutné zadat alespoň rozměry prostředí, na což je uživatel upozorněn dialogovým oknem.

MicInit Setup

Upozorňuje uživatele, aby nyní rozmístil mikrofony. Nachází se zde část formuláře **Microphone Setup**, konkrétně sekce s obecnými informacemi o souboru mikrofónů. Není zde umožněno specifikovat informace k jednotlivým mikrofónům, k tomu dojde později v **Microphone Setup**.

Speaker Setup

Umožní uživateli rozmístit reproduktor/y a zaznamenat informace o něm/nich. V horní části je sekce *Speaker Setup Info* obsahující pole pro vyplnění základních údajů o daném souboru reproduktorů (unikátní jméno, popis, datum). V části níže je možné přidat až **deset** různých reproduktorů do daného souboru. Každému reproduktoru je možné zadat řadu informací (pozice, směr, typ, popis, gain na zvukové kartě, gain na reproduktoru, hladina zvuku).

Mezi údaji specifikující směr a typ reproduktoru se nachází select box *Speaker Preset*, který po zvolení jedné z hodnot doplní následující položky podle zvolené hodnoty. Tyto hodnoty jsou obsaženy v souboru `webwizard/temp/presets/speakerPresets.csv` a jedná se o předem přidané hodnoty charakteristické pro často používané reproduktory. Pokud uživatel zvolí možnost *New Preset* a vyplní následující políčka, je mu umožněno tento „Preset“ uložit a později znovu použít.

Napravo od polí pro zadání pozice reproduktoru jsou posuvníky pro změnu typu požadovaných souřadnic. Uživatel může zvolit mezi **kartézskými** a **polárními** souřadnicemi, přičemž u obou lze vybrat, zda se jedná o souřadnice **absolutní**, měřené od počátku souřadné soustavy, nebo **relativní**, vztahující se k relativnímu bodu. V případě, že se jedná o první reproduktor v prvním souboru reproduktorů a Relativní bod nebyl zadán ve formuláři **Environment Setup**, není umožněno zadat relativní souřadnice, jelikož není relativní bod, ke kterému by se vztahovaly. Také u směru reproduktoru je možné měnit mezi relativními a absolutními údaji.

Napravo od posuvníků pro souřadnice a směr jsou tlačítka s ikonou žárovky, které uživateli poradí, co které údaje znamenají a v případě, že souřadnice ukazují mimo specifikované prostředí i upozornění, že jsou souřadnice mimo a o kolik.

Pokračování v procesu zadávání metadat je umožněno ve chvíli, kdy uživatel zadá jedny ze souřadnic a směrů každého z reproduktorů.

Microphone Setup

Je velmi podobný reproduktorům. Nahoře se nachází část *Microphone Setup Info*, která je v případě prvního souboru mikrofonů doplněna údaji vyplněnými na stránce **MicInit Setup**. Pod ní je sekce *Microphones of This Setup*, do které lze přidat až **šestnáct** mikrofonů. Každý mikrofon je opět specifikován souřadnicemi a směrem. Dále jsou zde políčka pro zadání umístění mikrofonu, paralelní osy prostředí, identifikátoru typu, upevnění mikrofonu, viditelnosti reproduktoru, gainu na zvukové kartě, vzorkovací frekvence, počtu bitů na vzorek, formát nahrávání a místo pro libovolné poznámky k danému mikrofonu.

Opět zde nacházíme select box - *Microphone presets*, který po zvolení jedné z možností doplní pole pod ním mimo pole pro poznámky. Tyto hodnoty jsou uloženy v souboru `webwizard/temp/presets/micPresets.csv` a jedná se o předem zadané hodnoty charakteristické pro často používané mikrofony. Pokud uživatel zvolí možnost *⟨New Preset⟩* a vyplní následující políčka (mimo *Notes*), je mu umožněno tento „Preset“ uložit a později znovu použít.

Stejným způsobem jako select boxy presetů fungují i select boxy pro zadání umístění mikrofonu a paralelní osy prostředí. Hodnoty pro select box „umístění“ se nachází v souboru `webwizard/temp/presets/micPlacing.csv`. Jedná se o nejčastěji používaná umístění, např. na nábytku, uvnitř nábytku, ve vzduchu. Select box pro paralelní osu prostředí obsahuje šest hodnot, které nelze měnit/doplňovat. Jedná se o všechny hranice prostředí (levá/pravá/přední/zadní zeď, strop, podlaha).

Napravo od souřadnic a směru mikrofonu jsou opět k dispozici posuvníky pro změnu chtěných souřadnic - *absolutní kartézské*, *relativní kartézské*, *absolutní polární* a *relativní polární*.

Finish

Stránka **Finish** je místem, kde může uživatel zjistit, co zapomněl vyplnit nebo vyplnil špatně. Tyto informace se nachází v dolní části stránky v sekcích *Environment Log*, *Speaker Log* a *Microphone Log*. Pokud se uživatel rozhodne, že chce některé z vypsanych chyb opravit, nebo jednoduše jen zkontrolovat vyplněná data, v sekci napravo jsou tlačítka odkazující na jednotlivé setupy, po jejichž stisknutí je otevřena nová záložka s podobným formulářem, jako při prvním vyplňování.

Hlavní součástí stránky Finish je vlevo nahoře, a to oblast s tlačítkem pro uložení souboru se všemi daty vyplněnými v průběhu průchodu webovou aplikací.

Zajímavým prvkem této stránky je nápověda viditelná po stisknutí tlačítka „HELP“ na navigační liště. Zde se nachází seznam věcí, které uživatel mohl přehlédnout či zapomenout. Je možné si jednotlivé položky „odškrtnout“ a zjistit, co chybí, nebo co bylo vyplněno špatně.

Edit Setup

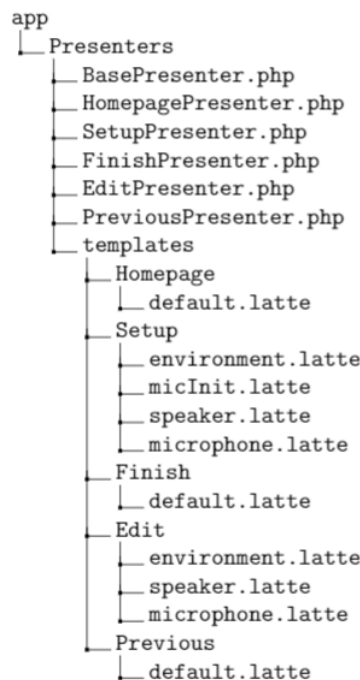
Formuláře pro editaci vyplněných dat jsou všechny spravovány třídou **EditPresenter**. Každý setup má vlastní stránku, kde můžeme měnit či doplnit jednotlivé položky formuláře. Pro jejich uložení je vespodu stránky tlačítko *Save*, které po stisknutí uživatele upozorní, že vyplněné hodnoty budou uloženy a stránka obnovena. Po uložení změn je možné stránku opustit zavřením její záložky.

5.3 Pohled na programovou část aplikace

Nette framework je využívající návrhový vzor *MVP* - *Model*, *View*, *Presenter*. „Model“ zajišťuje práci s daty a komunikuje pouze s Presenterem. „View“ je front-endovou vrstvou, která vykresluje požadovaná data a zobrazuje je uživateli. Vrstva „Presenter“ propojuje Model a View. Zpracovává požadavky, získává data z Modelu a posílá je View.

Díky tomuto návrhovému vzoru lze rozdělit aplikaci na logické celky a pro každý vytvořit vlastní Presenter. Presentery jsou soubory s příponou `.php`, Views mají koncovku `texttt.latte`.

Přechody na jednotlivé stránky umožňuje Nette pomocí identifikátoru `<presenter>:<view>` (např. `Homepage:default` nebo `Setup:microphone`). Struktura Presenterů a jejich Views je k vidění na Obrázku 5.3.



Obrázek 5.3: Struktura adresáře s Presentery a Views

5.3.1 BasePresenter

Abstraktní třída `BasePresenter` rozšiřuje Nette třídu `Nette\Application\UI\Presenter`. `BasePresenter` je rodičovskou třídou všech ostatních tříd presenterů implementovaných v

této aplikaci. Tvoří ji proměnná `$filePath` obsahující cestu k adresáři, se kterým aplikace pracuje a kam ukládá potřebné soubory (`/var/www/html/webwizard/temp/rec`).

V této třídě nalezneme také metodu `changeSettings`, která mění a ukládá hodnoty do souboru `settings.csv`. Tento soubor je zdrojem různých nastavení potřebných pro chod aplikace. Zpočátku k ukládání těchto hodnot sloužilo pole *superglobálních proměnných* `$_SESSION`. Tento způsob byl však zavrhnut z důvodu časového omezení uložení hodnot, jelikož nahrávání (tudíž i zapisování metadat) může trvat i několik dní a tyto hodnoty by za tuto dobu mohly vypršet a byl nahrazen přístupem ke zmíněnému souboru `settings.csv`.

5.3.2 HomepagePresenter

Třída `HomepagePresenter` zajišťuje přechod na stránky zobrazující metadata z předchozích nahrávání (metoda `handlePrevious`) a vytvoření krátkého formuláře, který uživateli umožní specifikovat název souboru, do kterého budou vyplněná metadata uložena a který si může na konci průchodu aplikací stáhnout.

Domovská stránka využívá šablonu `Homepage:default`. Tato šablona vykreslí stránku a zároveň vymaže nepotřebné údaje v *JavaScriptové* struktuře `localStorage`.

5.3.3 SetupPresenter

Soubor obsahující `SetupPresenter` je nejobsáhlejší složkou aplikace. Zahrnuje totiž ovládání čtyř různých views: *Setup:environment*, *Setup:micInit*, *Setup:speaker* a *Setup:microphone*.

Sestává se z třídy `EnvFormData`, která slouží jako úložiště pro hodnoty formuláře **Environment Setup**, `SpeakerFormData` obsahující hodnoty **Speaker Setup**, `MicFormData` sloužící pro ukládání hodnot formuláře **Microphone Setup** a třídy `SetupPresenter`, která obsahuje proměnné a metody pro všechny tyto formuláře.

`SpeakerFormData` a `MicFormData` se skládají z několika klasických proměnných pro popsání obecných informací o setupu a pole *multiplier*, jehož indexy reprezentují položky jedno určitého reproduktoru/mikrofonu.

Vyplnění jakékoliv ze stránek `Setup Presenteru` vytvoří soubor, do kterého jsou uložena získaná data ve formátu *csv*, tedy „typ, hodnota“. Tyto soubory jsou později sloučeny do jednoho a názvy daných hodnot jsou doplněny potřebnými prefixy. Při přechodu na Domovskou stránku jsou všechny tyto soubory odstraněny.

Environment

V metodách sloužících pro podporu formuláře `Environment Setup` nalezneme první vytváření komponenty pomocí frameworku `Nette`. Obdobné metody jsou k nalezení i u následujících `Setup`ů.

Nejprve je inicializována proměnná obsahující objekt třídy `Nette\Application\UI\Form`. Tento objekt je poté naplněn požadovanými poli pro jednotlivé informace o prostředí. Pole formuláře mohou být již při vytváření dle potřeby vyplněna.

Po vytvoření této komponenty je možné ji použít v *Latte* šabloně. *Latte* umožňuje používat speciální *tagy* pro vykreslování komponent přímo v *HTML* kódu. Formuláře jsou vykresleny ve formě tabulky, kde první sloupec tvoří element `label`, sloupec druhý element `input`. V případě, že je při definování komponenty přidána nějaká vlastnost (*description* apod.), může tabulka obsahovat více sloupců (v případě *description* je jím element `small`).

Příklad vykreslení komponenty v šabloně *environment.latte*:

```
// latte tagy jsou ohraničeny { }

{control EnvironmentForm}

// tímto způsobem je vykreslen celý formulář
```

Pokud je žádoucí změnit způsob vykreslení formuláře, je zapotřebí využít jiný postup. Tento postup byl využit při tvorbě této aplikace kvůli lepšímu definování vlastností jednotlivých HTML elementů:

```
// uvnitř oblasti ohraničené tagem {form} lze definovat vzhled formuláře
// cheme-li využít jednotlivé položky, musíme využít jejich identifikátoru

{form EnvironmentForm}
    <table>
        <tr>
            <th>{label EnvID}</th>
            <td>{input EnvID}</td>
        </tr>
        ...
        // případně
        <tr>
            <th><label n:name='EnvID' id='id' .../></th>
            <td><input n:name='EnvID' class='class' .../><td>
        </tr>
    </table>
{/form}
```

Příklad vytvoření komponenty Environment Form v *SetupPresenter.php*:

```
protected createComponentEnvironmentForm(): Form
{
    // vytvoření objektu Form
    $form = new Form;

    // obalení všech následujících položek do HTML elementu "fieldset"
    // až po další addGroup() nebo removeGroup()
    $form->addGroup('Basic Environment Info');

    // přidání položky určeného typu (integer, text, textarea...)
    // první argument je interní identifikátor, druhý obsah HTML elementu label
    $form->addInteger('EnvID', 'Environment ID');
    $form->addText('EnvName', 'Environment Name');
    ...
    $form->addTextArea('EnvDescription', 'Description');
    ...
}
```

```

// přidání tlačítka pro odeslání formuláře
$form->addSubmit('next', 'Continue');

// předvyplnění formuláře hodnotami ve struktuře $var
$form->setDefaults($var)

return $form;
}

```

Speaker

Komponenta **SpeakerForm** je vytvořena stejným způsobem jako **Environment Form**, avšak s drobným rozdílem. Formulář pro set reproduktorů může obsahovat více než pouze jeden reproduktor. Proto byla část formuláře pro jednotlivé mikrofony vytvořena za pomoci tzv. *multiplieru*. Zde byl využit *multiplier contributte/forms-multiplier*². Jeho využitím je umožněno po stisknutí tlačítka dynamicky rozšířit formulář o danou sadu polí pro specifikaci reproduktoru.

Některé informace o reproduktoru jsou typické pro určitý typ. Tyto informace mohou být automaticky vyplněny volbou možnosti ze select boxu *Speaker Preset*. Po zvolení některé z možností prohlížeč pošle serverové části aplikace *AJAXový* požadavek s využitím knihovny *jQuery*, který tato část vyhodnotí, načte příslušná data ze souboru obsahujícího jednotlivé presetu a zašle je ve formátu **JSON** zpět prohlížeči. Ten tato data vyplní do příslušných políček formuláře.

```

// klient (JavaScript)
$.ajax({
  method: 'get',
  url: {link LoadSpkPreset!},
  data: { data: hodnota_vybrane_moznosti },
  success: doplneni_prislusnych_elementu
});

// server (PHP)
$this->sendJSON((object)[
  'Type' => $type,
  'Description' => $description,
  'SndGain' => $sndGain,
  'SpkGain' => $spkGain,
  'NoiseLevel' => $noiseLevel,
]);

```

Klientská část obsahuje množství *JavaScriptových* funkcí pro ovládání stránky a kontrolu vstupu.

Funkce `change(el, id)` slouží právě ke zmíněné kontrole. Po změně hodnoty souřadnic funkce vypočítá, zda se zadané souřadnice nenachází mimo rozměry místnosti. Pokud ano, je políčko s danou hodnotou červeně zvýrazněno a po stisknutí žárovky vpravo od něho uživatel

²<https://github.com/contributte/forms-multiplier>

může zjistit bližší informace (o kolik je daná souřadnice mimo). Současně se informace o „špatných“ souřadnicích uloží do struktury `localStorage`, která je později využita pro zvýraznění daných políček při editaci daných setupů.

Výchozí souřadnice zobrazené ve formuláři jsou **absolutní kartézské**. Chce-li uživatel zadat jiný typ souřadnic, může napravo od oblasti se souřadnicemi využít posuvníků, které dané souřadnice změní. Jestliže nebyl v `Environment Setup` zadán relativní bod, nelze zadat relativní souřadnice.

Po stisknutí tlačítka, které uživatele přesune na jinou stránku je provedena kontrola, zda byly u všech reproduktorů daného setupu vyplněny některé ze souřadnic a informace o směru reproduktoru. Pokud ne, je zobrazeno plovoucí okno upozorňující, že bez vyplnění těchto položek nelze pokračovat dále v průchodu aplikací. Jsou-li položky vyplněny a relativní bod nebyl specifikován v **`Environment Setup`**, je uživatel dialogovým oknem upozorněn, že pro upřesnění relativního bodu budou využity souřadnice prvního reproduktoru. Toto platí pouze, pokud je právě vyplňovaným setupem první set reproduktorů.

Při opuštění stránky je AJAXovým požadavkem uložen text specifikující nesrovnalosti ve vyplněných datech (zda byla některá políčka ponechána prázdná či jaké souřadnice ukazují mimo místnost). Tento text je uložen do příslušného souboru pro pozdější zobrazení na finální stránce aplikace.

Microphone a MicInit

Tyto části sdílí stejné metody třídy `SetupPresenter`, ovšem liší se ve svých šablonách. Formulář **`MicInit Setup`** je vlastně „ořezaný“ formulář **`Microphone Setup`**; je z něj využita pouze sekce s obecnými informacemi o setupu.

Komponenty těchto formulářů jsou vytvořeny metodou `createComponentMicForm` a liší se od formuláře pro reproduktor pouze v požadovaných hodnotách. Jelikož jsou oba formuláře prakticky stejné, i další metody se nijak markantně neliší od metod pro reproduktory.

Rozdílem v klientské části je možnost výběru relativních souřadnic. V tento moment byl totiž relativní bod zadán buď v **`Environment Setup`**, nebo byl použit první reproduktor z prvního setu reproduktorů.

Opět zde nacházíme i kontrolu jednotlivých souřadnic a případné uložení textu o nesrovnalostech.

5.3.4 FinishPresenter

Tato třída se skládá z metod, které ovládají stránku `Finish`. Ta je (mimo editaci) poslední zastávkou průchodu webovou aplikací. Hlavní složkou této stránky je část vlevo nahoře. Obsahuje tlačítko, po jehož stisknutí jsou soubory s vyplněnými daty sloučeny do jednoho s názvem, který specifikoval uživatel (případně je využit název `Prostředí`, nebo `rec_metadata_ROK-MĚSÍC-DEN_HODINY:MINUTY`). Současně se sloučením jsou k názvům jednotlivých hodnot doplněny potřebné prefixy. Poté dojde ke stažení souboru na uživatělv počítač.

Spodní sekci stránky tvoří tři oblasti, každá pro jeden ze setupů (**`Environment`**, **`Speaker`**, **`Microphone`**). Tyto oblasti obsahují informativní texty o zadáných souřadnicích, které ukazují mimo místnost, a o polích, které nebyly uživatelem vyplněny. Jestliže nejsou k dispozici soubory s těmito texty, považuje aplikace vyplněné hodnoty za správné a místo daných textů zobrazí přivětivou zprávu `All good!`.

Napravo od hlavní části stránky s tlačítkem *Download* je seznam vyplněných setupů. V tuto chvíli může uživatel po kliknutí na libovolný setup změnit, doplnit či vymazat libovolné hodnoty daného setupu.

5.3.5 EditPresenter

Pro editaci jakéhokoliv ze setupů byla implementována třída **EditPresenter**. Její struktura i metody jsou velmi podobné presenterům ovládajícím původní setupy. Rozdílem je tvoření komponenty formuláře - nyní je známý počet reproduktorů/mikrofonů v daném setupu, není tudíž zapotřebí nadále využívat *multiplier*. Pro tvorbu jednotlivých složek stačí prostý **for** cyklus. Přidávání dalších reproduktorů či mikrofonů není umožněno, jelikož nahrávání těchto setupů již proběhlo.

Po volbě setupu, který chce uživatel editovat, je otevřena nová záložka obsahující daný formulář.

Pro uložení změněných, smazaných či doplněných hodnot je pod formulářem umístěno tlačítko *Save*, které po stisknutí uživatele upozorní, že budou změny uloženy a stránka bude znovu načtena. Poté může uživatel opustit záložku jejím zavřením, nebo stiskem tlačítka *Quit Editing*, které způsobí totéž.

5.3.6 PreviousPresenter

Tento presenter slouží k zobrazení metadat o předchozích nahráváníích. K přístupu na stránku s nimi slouží odkazy na Domovské stránce, v sekci v pravé části stránky.

Tyto odkazy jsou tvořeny prozkoumáním adresáře `webwizard/temp/rec/prev`, kam jsou uloženy soubory vytvořené na poslední stránce průchodu aplikací, stránce **Finish**.

Údaje jsou zobrazeny v přehledných tabulkách ke každému ze setupů. Tyto tabulky je pro větší přehlednost možné sbalit či rozbalit. Není možné je editovat, stránka slouží pouze k přehlednému zobrazení metadat, aby uživatel nemusel pracně procházet a snažit se pochopit jednotlivé soubory.

5.4 Testování aplikace

Testování probíhalo manuálně, konkrétně opakovanými průchody aplikací, vyplňováním jednotlivých formulářů a simulací chybných vstupů.

5.4.1 Výstup aplikace

Jelikož je výsledkem průchodu aplikací soubor obsahující zapsaná data ve formátu *csv*, bylo toto testování velmi přímočaré. Mezi soubory dodanými společně se zadáním práce byl soubor, který obsahuje několik různých naměřených místností nacházejících se na FITu i mimo něj. Účelem bylo zjistit, zda po vyplnění hodnot jako obsažených ve zmíněném souboru dostaneme totožný soubor.

Toto testování odhalilo, že aplikace svůj hlavní úkol splňuje správně. Vytvořený soubor odpovídá dodaným datům. Ta sice obsahovala i popisky jednotlivých položek, tato skutečnost však byla implementována přímo do webové aplikace.

5.4.2 Funkcionalita

Další testování spočívalo ve zkoumání, zda po zadání chybných vstupů, nebo špatným zacházením se stránkou, nedojde k neočekávaným chybám aplikace. I zde se ve většině případů setkáme s očekávanými situacemi. Po zadání souřadnic ukazujících mimo místnost jsou tyto souřadnice správně červeně vybarveny a informace o špatné pozici promítnuty do oblasti s radami k jednotlivým souřadnicím. Polární souřadnice jsou správně převedeny na kartézské a relativní souřadnice se odkazují na relativní bod.

5.5 Známé problémy aplikace

Součástí webové aplikace je, bohužel, i několik chyb. Tyto chyby byly odhaleny při testování aplikace, avšak z časových důvodů je nebylo možné vyřešit. Důvodem byla nastalá vládní opatření, zejména zákaz volného pohybu mezi okresy. Jelikož nejsem z Brna, kde mám zajištěné ubytování na kolejích, a zmíněná opatření přišla rychle, neměl jsem nějakou dobu k dispozici zvukovou aparaturu ani pracovní laptop, na kterém má práce probíhala.

Jedním ze známých nedostatků jsou logy vytvářené při nevyplnění polí formuláře, či při souřadnicích ukazujících mimo místnost. Tyto logy jsou bez problémů při používání prohlížeče *Google Chrome*, avšak u jiných prohlížečů může dojít k nevytvoření souborů s textem o těchto chybách. Navíc položky, které uživatel opraví v Editaci Setupů, nejsou odstraněny, uživatel si tudíž musí pamatovat, co již opravil a co ne.

Problémem může být také nemožnost návratu zpět na předchozí setup. Toto by neměl být problém nijak markantní, jelikož editace setupů je možná z poslední stránky průchodu aplikací. Pokud však uživatel využije v prohlížeči tlačítko „Zpět“, může dojít k nekonzistenci dat.

Kapitola 6

Závěr

Cílem práce bylo zejména vytvořit webovou aplikaci pro usnadnění zápisu metadat o nahrávání. Současně s tím byl sepsán i podrobný návod pro zacházení se zvukovým systémem a s ním spojeným softwarem.

Seznámil jsem se s dodaným hardware i software. Pro HW ani SW dříve neexistovala v podstatě žádná dokumentace či návod k používání. Důkladně jsem proto prostudoval práci pana Martina Lacha, která mi pomohla pochopit jak fungují program pro nahrávání a odesílání dat Streamer, program pro příjem a ukládání přijatých dat Receiver a vizualizační a řídicí webová aplikace.

Výsledkem práce jsou opravy některých chyb dodaného SW, přidání jednoduché funkcionality, návod k používání HW i SW a lokální webová aplikace sloužící pro usnadnění a pomoc při zaznamenávání metadat o nahrávání.

Vytvořená aplikace s názvem Webwizard funguje dle představ a významně ulehčuje uživateli práci se zápisem údajů o prostředí, kde probíhá nahrávání audio dat, i o použitých reproduktorech a mikrofonech. Její zdrojové kódy jsou dostatečně komentované pro případné úpravy či rozšíření.

Tato aplikace nejenže uživateli pomůže vytvořit přehledný soubor s danými metadaty, rovněž určí pořadí kroků, které musí uživatel vykonat pro získání dat z měření. Pokud si uživatel s některými údaji neví rady, aplikace mu napoví.

Při konzultacích s vedoucím práce jsme se shodli, že při pokračování v navazujícím magisterském studiu bych mohl mít možnost aplikaci případně doladit či rozšířit o další funkcionality, aby co možná nejvíce vyhovovala potřebám jejích uživatelů na FIT VUT.

Literatura

- [1] GROUP, T. P. *PHP: Hypertext Preprocessor* [online]. The PHP Group, 2001 - 2021 [cit. 2021-05-05]. Dostupné z: <https://www.php.net/>.
- [2] GRUDL, D., ČERNÝ, H., HŮLA, M. a ŠULC, M. *Latte - nejbezpečnější & opravdu intuitivní šablony pro PHP* [online]. Nette Foundation, 2008, 2021 [cit. 2021-05-05]. Dostupné z: <https://latte.nette.org/cs/>.
- [3] GRUDL, D., ČERNÝ, H., HŮLA, M. a ŠULC, M. *Nette - Pohodlný a bezpečný vývoj webových aplikací v PHP* [online]. Nette Foundation, 2008, 2021 [cit. 2021-05-05]. Dostupné z: <https://nette.org/cs/>.
- [4] LACH, M. *Vizualizace parametrů mnohakanálového zvukového systému v internetovém prohlížeči*. Brno, CZ, 2020. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Dostupné z: <https://www.fit.vut.cz/study/thesis/23440/>.

Příloha A

Obsah přiloženého paměťového média

A.1 Software

A.1.1 Archiv *webwizard.zip* obsahující vytvořenou webovou aplikaci

A.1.2 Archiv *zvukovka.zip* s upravenými soubory pro nahrávací zařízení

A.2 Soubor *readme*

A.3 Text Bakalářské práce ve formátu PDF

A.4 Archiv se zdrojovým kódem Bakalářské práce

A.5 Videoprezentace práce

A.6 Plakát

Příloha B

Manuál

B.1 Webwizard

Pro úspěšnou instalaci a chod webové aplikace Webwizard je zapotřebí jazyk PHP verze 8.0 nebo novější a webový server (Apache2, Nginx).

Postup při instalaci aplikace je následující:

- Instalace HTTP webového serveru Apache2

```
sudo apt-get update
sudo apt install apache2
```
- Instalace jazyka PHP 8.0 pro server Apache2

```
sudo add-apt-repository ppa:ondrej/php
sudo apt install php8.0 libapache2-mod-php8.0
```
- Nakopírování kořenového adresáře aplikace

```
cd /var/www/html
sudo chmod -R 777 /var/www/html
do /var/www/html nakopírujte složku „webwizard“
sudo chmod -R 777 /var/www/html/webwizard
```
- Konfigurace webového serveru Apache2

```
sudo nano /etc/apache2/apache2.conf
```

v sekci <Directory /var/www> je třeba změnit hodnotu
None u AllowOverride na All

```
sudo a2enmod rewrite
sudo systemctl restart apache2
```

Nyní je aplikace připravena k použití. Do internetového prohlížeče stačí zadat URL 127.0.0.1/webwizard/www nebo localhost/webwizard/www.

B.2 Upravené soubory nahrávacího systému

Soubory obsažené v adresáři „Zvukovka“ na paměťovém médiu stačí nakopírovat do jednotlivých složek původního software. Struktura adresáře odpovídá struktuře původního.